

WebView的使用

在Flutter中显示webView比较常见，一般使用的功能就是和JS的交互，现在使用WebView主流的两个插件：

[flutter_webview_plugin](#)，[webview_flutter](#)(官方推荐)。下面以flutter_webview_plugin使用为例做介绍。

引入方式

- 在pubspec.yaml 中导入 flutter_webview_plugin 插件；
- 执行flutter pub get获取插件；

使用方式

- 在使用的文件中添加引入

```
import 'package:flutter_webview_plugin/flutter_webview_plugin.dart';
```

- 属性

```
const WebviewScaffold({
  Key key,
  this.appBar, //类似于Scaffold中的头部widget
  @required this.url, //加载的url
  this.headers, //
  this.withJavascript, //是否允许执行js代码
  this.clearCache, //
  this.clearCookies, //
  this.enableAppScheme, //
  this.userAgent, //
  this.primary = true, //
  this.persistentFooterButtons, //
  this.bottomNavigationBar, //
  this.withZoom, //是否允许网页缩放
  this.withLocalStorage, //是否允许LocalStorage
  this.withLocalUrl, //
  this.scrollBar, //是否显示滚动条
  this.supportMultipleWindows, //
  this.appCacheEnabled, //
  this.hidden = false, //
  this.initialChild, //
  this.allowFileURLs, //
  this.resizeToAvoidBottomInset = false, //
  this.invalidUrlRegex, //
  this.geolocationEnabled, //
})
```

```
})
```

- 监听webView的动作: `FlutterWebViewPlugin` 插件提供一个链接到唯一webview的单一实例, 这样就可以在app中的任何地方控制webview

```
// 监听url地址改变事件
flutterWebViewPlugin.onUrlChanged.listen((String url) {
});
// 监听web页加载状态
flutterWebViewPlugin.onStateChanged.listen((WebViewStateChanged
webViewState) {
});
```

```
// 监听页面滚动事件
flutterWebViewPlugin.onScrollYChanged.listen((double offsetY) {
});
flutterWebViewPlugin.onScrollXChanged.listen((double offsetX) {
});
```

```
// 隐藏webview
flutterWebViewPlugin.launch(url, hidden: true);
// 关闭webview
flutterWebViewPlugin.close();
```

关于JS交互, 可以阅读文章[点击这里](#)

注意点

- webview并不存在于widget树中, 所以你不能在webview中使用如snackbars, dialogs等
- 在iOS设置中需要做如下添加

```
<key>io.flutter.embedded_views_preview</key>
<string>YES</string>
```

- 如果网页是http链接, 解决方案是: <https://juejin.im/post/6844903908087693319>